Programmable Filament: Printed Filaments for Multi-material 3D Printing

Haruki Takahashi
Meiji University
Tokyo, Japan
haruki@meiji.ac.jp

Parinya Punpongsanon
Osaka University
Osaka, Japan
parinya@sys.es.osaka-u.ac.jp

Jeeeun Kim
Texas A&M University
College Station, Texas
jeeeun.kim@tamu.edu

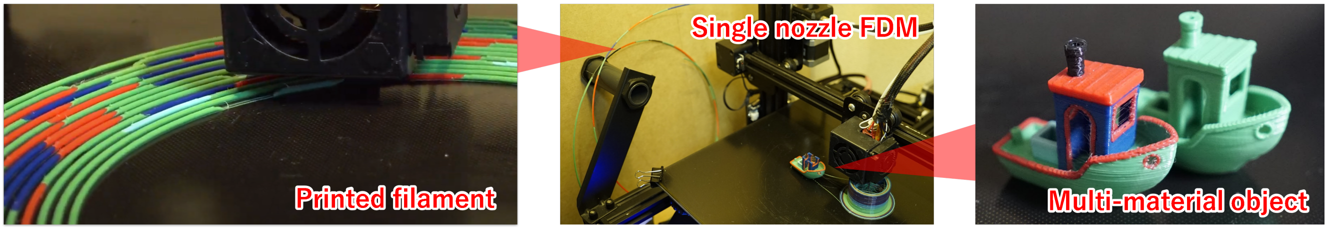


Figure 1. Programmable Filament is a novel 3D printing technique that enables users to 3D print an object with multiple materials using an FDM printer without any hardware modification. (From left to right) First, users generate a filament that contains multiple materials, to feed into the extruder, then 3D print an object in full color.

ABSTRACT

From full-color objects to functional capacitive artifacts, 3D printing multi-materials became essential to broaden the application areas of digital fabrication. We present Programmable Filament, a novel technique that enables multi-material printing using a commodity FDM 3D printer, requiring no hardware upgrades. Our technique builds upon an existing printing technique in which multiple filament segments are printed and spliced into a single threaded filament. We propose an end-to-end pipeline for 3D printing an object in multi-materials, with an introduction of the design systems for end-users. Optimized for low-cost, single-nozzle FDM 3D printers, the system is built upon our computational analysis and experiments to enhance its validity over various printers and materials to design and produce a programmable filament. Finally, we discuss application examples and speculate the future with its potential, such as custom filament manufacturing on-demand.

Author Keywords

3D printing; fused deposition modeling; programmable matters; multiple materials.

CCS Concepts

•Human-centered computing → Interactive systems and tools;

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

UIST '20, October 20–23, 2020, Virtual Event, USA
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7514-6/20/10 ...$15.00.
https://doi.org/10.1145/3379337.3415863

INTRODUCTION

With recent advances in consumer grade 3D printers, end users without professional skills are able to fabricate various objects from a smartphone case to custom mechanical tools. Along with the progress of machinery, there has been a growing desire for multi-color and multi-material 3D printing using fused deposition modeling (FDM), not only to get an artifact aesthetically appealing but also to enable the production of an object with properties not found in nature, i.e., metamaterials, even when faced with a limited choice of materials (e.g.,[1, 2, 9]). However, the materials used in low-cost 3D printers are fairly limited due to the form factor and affordability. Commodity FDM machines are therefore often equipped with only a single, or dual extruder at most. Furthermore, multi-material printing with a dual extruder is not simple, and incurs many practical issues such as oozing, stringing, extruder-object collisions, first layer shifting, and an under-extrusion caused by drizzling, occurring during the switching between extruders. Despite advanced techniques to tackle these challenges, such as FDM 3D printers with a built-in ink cartridge [41], inkjet-based 3D printing [35, 40] and a 3D printer that can seamlessly switch different materials within a single nozzle [33], these techniques are generally inaccessible to average usera because they often require professional skills to manipulate the materials, machines, and processes.

In this work, we introduce *Programmable Filament*, a novel 3D printing technique that enables users to print an object with multiple colors and materials, using an FDM without any hardware updates. Inspired by the idea of 3D printing a thin wire resembling a filament by Instructables user DasMia [3], we expand on the concept to fabricate a programmable filament connecting several segments of various materials into a single filament based on the user's specifications (referred to as *printed filament*). We demonstrate that the printed filament can be used in the same way as a conventional filament, i.e., extrude through a standard nozzle, requiring no hardware modification. We further explore a calibration technique used to minimize the shifting or mixture of the colors and/or materials when switching between segments. Our approach addresses challenges in dual printing listed above, as we only use a single nozzle to print numerous types and colors of filament.

We first review relevant work in multi-material printing and advanced techniques to extend the capabilities of desktop printers. We start an introduction of programmable filament technique with a step-by-step process of printing filaments and analyze obtained characteristics. We follow up with a description of experiments conducted to enhance the quality of the printed filaments. Integrating findings from these empirical experiments, we present a filament design system for end-users, inviting them to program a filament based on 3D objects or images as input to create a ready-to-print filament. We conclude with a range of application examples demonstrating our approach, and discussion on the remaining challenges and possibilities for future research direction.

RELATED WORK

Our work is related to the 3D printing in multiple materials, application areas, and various techniques used to overcome the limitations found in FDM. Our technique takes direct G-code manipulation approach to yield non-standard effects.

Increasing Needs in Multi-material Printing

Since the commodity FDM 3D printer has been introduced to the consumer market, significant advances have been made in materials suitable for these machines. As plastic melts at relatively low temperatures, filament manufacturers started to vary its properties by simply infusing special pigments into standard plastic (e.g., PLA), keeping its nature of easy storing and safety. Single threaded full-color filament (an example is available at [31]) seemed promising, despite its inability to locate each color at the selected parts. The ability to make filaments with special pigments (e.g., a graphite, ferromagnetic, or carbon fiber) inspired researchers in the HCI community to create functional 3D objects, such as capacitive objects [4, 29], magnetically encoded wireless devices [10, 11], or identifiable artifacts with barcode [21]. Exploiting flexible thermoplastic such as TPU, through the different buckling behaviors of heterogeneous materials when attached together, the transformation of a flat sheet into a sophisticated 3D shape has become available for desktop 3D printers [2]. Use of water-soluble material with regular plastic further expanded application scenarios of 3D printed artifacts, such as optically identifiable object [18], object with wash-away assembly keys [23] and more. Nonetheless, these approaches have mostly been possible with advanced hardware, which still presents numerous practical issues, listed above. Moreover, additional potential has not been revealed any further, largely due to only two materials available at a time.

Techniques for Fabricating Multi-color/material Objects

Many hardware-based attempts have been made to tackle the 3D printing of multiple colors and materials. To make full use of FDM in multi-color 3D printing, practitioners have been investigating various hardware-oriented approaches, including four extruder headers [15] and the use of external hardware to merge multiple materials before fusing them into a single nozzle for full-color object printing [14]. Song et al. proposed a modified print header to allow multiple filaments to be extruded through a single nozzle [34]. These multiple colors are mixed and provide almost full color gamut, according to the programmed melting ratio. *Jubilee* is an open-source toolkit to enable automated tool-changing for material explorations [39]. For typical home users, researchers also developed a low-cost technique, by manually changing a filament with pause-and-resume [20]. However, multi-material printing remains challenging, even with hardware upgrades, the number of materials are still limited, and users are limited with the number of extruders they can apply. Although *CleanColor* [8] helped resolve the issue of mixed colors appearing in dual-nozzle printing, users are typically occupied with tedious calibrations and required maintenance tasks. Alternatively, an advanced additive-manufacturing technique has been investigated, such as inkjet 3D printing with special materials [24, 30, 38, 32]. Post-processing is another alternative, painting in full colors, spraying electro-magnetic [44] or color-changeable paint [12]. However, hardware upgrades prevent non-experts from being able to afford these solutions, and additional steps for post-processing makes it hard to apply them in practice.

Direct G-code Manipulation to Enhance FDM Capability

Recent works have begun to use of G-code to directly control the mechanisms of machinery so to yield non-traditional effects from FDM. G-code is a series of commands widely applied to numerically controlled machines (CNC), commands to control movement axis (G1), speed (F), extrusion amount (E), and printing temperature (M109) are generated by interpreting the user's choice of slicing parameters. Although a set of recommended parameters for specific machine types and materials have been proposed to guarantee the print quality, adjusting values or command generation mechanisms yields special effects that are unlikely to achieve through standard settings. New ways to overcome the existing limitations in the current form factors of consumer-grade FDM machines [28] have been explored, e.g., *3D printed hair* proposes extending a stringing phenomenon of extruded materials for fabricating fine hair-like textures [17]. *WirePrint* allows diagonal z-movements of a delta style printer to construct sparse wireframe meshes so reduce the printing time [22]. Controlling the height of the nozzle as well as the amount of material extrusion, expressive textures such as a fluffy surface can be created [37]. *3D printed fabric* is an additional technique enabling the weaving movements of the header, enabling users to create a flexible fabric out of rigid plastic [36]. *Compositional 3D Printing* also enables a pause-to-insert action and drooping effects, resulting in the explorative, intervention-oriented techniques for design-by-print [13]. *Dual-color mixing* [25] and *3D hatching* [16] use G-codes at low level to express the tone of 3D prints by fine-tuning color layering. Our technique also employs a similar approach to print a programmable filament, and applies the printed filament to obtain a new material/coloring capability in FDM. In the remainder of the paper, we refer to an FDM 3D printer simply as a 3D printer.

PROGRAMMABLE FILAMENT

We first describe an overview of the printing a single threaded filament printed by splicing multiple segments.

Printing Filament: Process

Our approach is inspired by a craft technique that splices segments of multiple resource filaments to fabricate a single resulting filament. By printing a long spiral consisting of several segments, the resulting filament can be used as regular material similar to the off-the-shelf filament for 3D printing [3]. We extend this by directly manipulating G-codes to control 3D printer mechanisms and create a programmable filament. Even without a fixed *.stl* file of a spiral introduced in this work with limited control, we can fine-tune the characteristics such as the thickness and roundness of the edges, the number of materials to be spliced, adding a special structure to join each segment by directly manipulating the G-code.

Figure 2 shows an exemplary procedure of a printing filament—printing a single threaded, three-colored filament in order of red, green and blue. First, the 3D printer prints a brim that helps adhere the first layer to the printing bed, and then starts printing all red segments onto the brim. The endpoints of each segment are diagonally cut (Figure 2a), to avoid a collision with the nozzle when printing an adjacent segment for the second color. After completing the printing, all red segments in the spiral, the 3D printer *pauses* and moves its header to the XY-home of the build area, to be ready for exchanging filament (Figure 2b). While pausing, the user manually exchanges the currently plugged filament with a green filament. We control the printer mechanisms for user-intervention by inserting G-code commands in the middle to pause the printing (M0), which notifies the user with a beep sound (M300). The *M0* helps pause the printing without cooling the nozzle or bed while waiting. If the 3D printer has an LCD, it displays a short message instruction letting users know the right material to plug in. Resuming the printing process initiates an extrusion of the green filament, by purging the remaining materials in the feeder and nozzle. The green segments are then ready, allowing the next segments to be printed. If the segment starts from the endpoint of the prior segment, the printer prints a green part with an upside-down diagonal cut to match the contact surface of the prior segment (Figure 2c). After printing all green segments, the 3D printer pauses again, and iterates the above process (Figure2d). Finally, the 3D printer adds stitches that patch the joint of the segments on top (Figure 2e). The printed filament is now ready to be plugged in for 3D printing in the same fashion as an off-the-shelf filament.

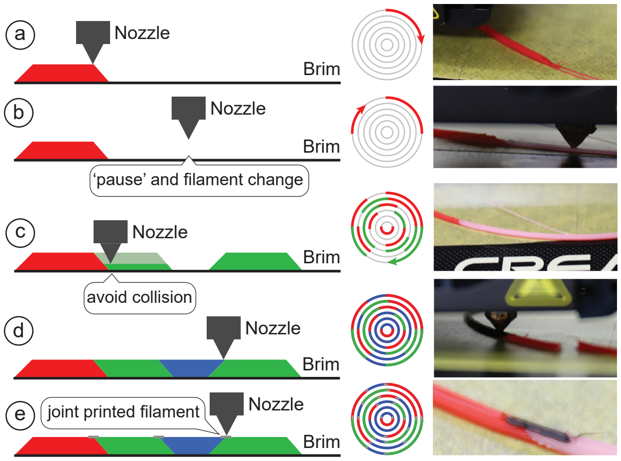


Figure 2. Printing procedure of a filament: (a) Printing starts with one color, (b) it pauses upon completion of printing all segments, allowing the user to change the material. (c-d) The 3D printer prints the remaining segments avoiding collision with prior segments, (e) then prints stitches to join adjacent segments.

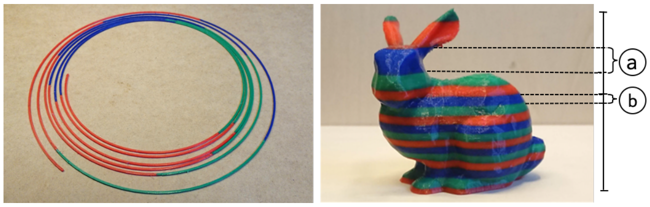


Figure 3. An exemplary printed filament and object printed using it: The layers of (a) and (b) are printed in the same segment length (200 *mm*), but appear with different layer counts.

Programming Filament: Computations

Our technique makes use of a programmable filament to create an on-demand filament to be used as a regular off-the-shelf filament (Figure 3). We envision a unique usage of a printed filament that empowers users to control the extrusion of the printed filament according to the printing path. For example, the height of each stripe differs when printed by the segments in the same length. With each color-segment in the same length in the thread, only approximately 10+ layers of a blue segment in the body can be printed (Figure 3b), whereas it can print approximately 30+ layers on its face (Figure 3a) due to the discrepancy of the surface area on each layer. Here, we describe how to compute the relationship between the amount of material required to print a series of segments from the resource filament, as well as the length of each segment in the resulting spiral path. We then describe the calibration process.

Measuring Source Filament Length for Resulting Filament

The amount of extrusion is determined by the E parameter value in millimeter in the G1 command (we refer this as *e-mm* in unit for E-push length, and how long of a filament is fed during the movement distance that we refer this as *t-mm* in unit for travel distance). For example, when a 3D printer receives the command *G1 X10 E1* at the origin $(x,y) = (0,0)$, the header moves to $(10,0)$ (X10) while extruding 1 *e-mm* of material (E1). The E value is calculated from the form factors of each 3D printer as follows:

$$E[mm] = \frac{W_{nozzle} \times H_{layer} \times L_{printing}}{C_{filament}} \quad (1)$$

Assume a nozzle diameter W_{nozzle} of 0.4 *mm* (the most common size), a layer height H_{layer} of 0.4 *mm*, a printing length $L_{printing}$ of 1 *mm* (unit length), and a cross-sectional area of a filament $C_{filament}$ of 2.405 mm^2 using a 1.75 *mm* diameter filament. The computed E value is 0.0665 *mm* in this case. By transforming Equation 1 to compute $L_{printing}$ from the E value; for the same case, we can print a 15.033 *t-mm* long line ($L_{printing}$) on the bed using 1 *e-mm* of the resource filament (E).

Using this equation and an input 3D model, we can compute the length of each segment in a printed filament. If the 3D model is already sliced into G-code, we only need to add the E values in the G1 commands. We can then control the amount of filament to be extruded under the following two conditions: The 3D printer (1) can extrude a precise amount of material and (2) can extrude the materials along a pre-generated path that enables the 3D printer to correctly switch the materials (see the implementation section).

Calibration for Actual Use of Printed Filament

We explore a calibration technique to meet the first condition above by calibrating the stepper motor used to feed a filament into the extruder. Our calibration is based on a Marlin[1] open-source system, which calculates the rotation distance from the number of steps where the stepper motor moves to extrude 1 *mm* of filament (E1), i.e., in $steps/mm$, called an *E-step*. In mechanical terms, this value is known by the characteristics of the 3D printer's form factors, such as the motor's step angle, belt pitch, and number of teeth in each gear. The E-step needs to be calibrated, allowing the 3D printer to extrude an accurate amount of material.

We conducted a manual calibration, following a well-known calibration technique described by E3D in [26]. In short, we mark the filament to be inserted into the extruder end at 100 and 120 *mm*, and extrude a 100 *mm* (E100) filament. After the extrusion, we measure the actual length of the filament that has been extruded and the remaining amount from the mark at 100 *mm*. If the 3D printer extruded too much passing 100 *mm* mark, we can measure it using the mark at 120 *mm*. We then compute a new E-step value based on the extrusion amount, $E_{new} = E_{current} \times 100/\ L_{extruded}$, and so overwrite the firmware setting with this value.

Printed Filament: Analysis

Here, we describe the characteristics of the printed filaments and associated constraints that need to consider when used for 3D printing.

#1 Maximum Length: The maximum length of a printed filament is highly dependent on the build area given the type of 3D printer. With a 300×300 *mm* bed size, a printed filament in a spiral shape can be approximately 20-m long. The innermost radius of the spiral should be sufficiently large to prevent breakage while plugging it into the feeder, upon completion of the printing filament. We set it to 30 *mm* by referring to a commercial spool size.

#2 Density: Similar to the existing filament splicing technique [3], the cross-sectional area of our printed filament is hexagonal (Figure 4a). Therefore, when this filament is used in actual 3D printing, the amount of material to be extruded may not be the optimum value which is measured for a full cylindrical shape (dotted lines). If an under- or over-extrusion occurs when using it as a regular filament, the flow rate needs to be calibrated [27].

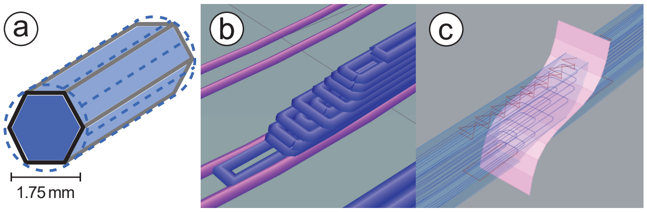


Figure 4. (a) The cross-sectional area of printed filaments is hexagonal and (b, c) the endpoints of each segment are cut by a wavy-cutting plane.

#3 Fabrication Cost: There are three types of fabrication costs for a printed filament: the printing time, amount of material, and manual labor. The costs differ according to the length and number of materials used. As an example, a 1-m long printed filament that consists of 20 segments (50 *mm* per segment) in 3 types of materials takes approximately 36 *min.* to print (excluding the time required for material replacement). Note that we empirically set our printing speed to 1,200 mm/min. and the movement speed to 9,000 mm/min. To change the materials, we estimate that it takes approximately 3–5 *min.* from the removal of a material from the nozzle to inserting the next material, purging, and resuming to the original position. Because we fabricate using resource filaments, the amount of material required to print a filament is almost equal to the sum of their lengths. The purging requires an extra amount of material to replace the filament without residue, and we currently set a 50 *mm* filament for purging. Finally, some manual effort is required for removing and inserting a material. Because the 3D printer informs the user each time the material is changed, the number of user interventions for an exchange is equal to the number of materials minus 1 (i.e., a single thread consisting of 5 colors requires 4 exchanges).

#4 Connectivity between Segments: The endpoints of one segment in contact with the adjacent segments are diagonally cut with a wavy shaped surface to increase the contact area (Figure 4b, c). An additional layer that covers two materials across is printed on top and on the bottom to enhance the binding (we refer to this layer as a stitch). The stitch itself and heat added by having the hot nozzle passes along when printing the stitch on top contributes to strengthen the adhesion of two heterogeneous segments. We confirmed that various colored PLAs can adhere to each other tightly. However, each material shows different characteristics (e.g., viscosity and stiffness), and the binding strength between two materials may differ. We will further extend the experiment to validate the effects, particularly with various material combinations such as TPU + PLA in the following section.

PROPERTIES OF PRINTED FILAMENT

We conducted several experiments to explore the new characteristics occurred to the printed filaments, validity across various machines and material types, and developed techniques to mitigate any negative effects.

Validation on Various Machines for Replicability

One main goal of this work is to overcome the challenges in multi-color/material printing that generally occur when using a consumer grade FDM 3D printer, particularly with a single nozzle. Hence, we conducted several experiments using various single-nozzle platforms. We also extended our experiments to verify the printability over numerous variations of

---

[1] Marlin Firmware (`https://marlinfw.org/`)

FDM. All tests were iterated using four different 3D printers, namely, a Creality CR-10S, Prusa MK teacup, Creality Ender 3, and NWA3D. These four printers come with similar configurations but different gear setups. We also tested a Geeetech A20M (with a single nozzle and dual extruders) and a Creality CR-10S Pro (with dual gear extruders).

Validation on Material Adhesion for Generalizability

We examine whether our approach applies to various material combinations, such as PLA, ABS, TPU (flexible material), conductive PLA, nylon, and PVA (water-soluble filament), obtained from several brands for generalizability. We purposefully chose these sets of materials to consider their future possibilities in fabricating functional objects using produced programmable filaments. As previously identified by Mosaic Palette, a hardware attachment manufacturer of splicing materials [14], various combinations of materials affect the quality of resulting filament. Here, we describe our results in generating a short filament consisting of two types of material segments and present the empirical findings.

We first confirmed that all materials can form a filament shape in a spiral when printed. It is clearly harder to connect two heterogeneous materials than the same material in different colors. To ensure stability, it is necessary to adjust the printing parameters (printing temperature, flow rate, and printing speed) when switching between two types of materials, e.g., ABS from PLA, following the recommended settings from the manufacturers. For instance, to print TPU segments after completing PLA segments, the printing temperature of the nozzle must be increased, and slower printing speed is recommended. The type of combination and adhesion strength also affect the feeding of the printed filament when the resulting filament is used. As the torsional force is applied to the printed filament when fed, caution must be used to determine the insertion direction. Furthermore, we found there are various extrusion conditions in the printed filament. For example, for a printed filament consisting of TPU and PLA, there are four combinations of which material pushes out which one, and the TPU is too soft to push a PLA segment toward the nozzle. Even if we confirm that all materials listed above can be used to create a printed filament in any combinations, certain joints (but not all) will break if bent harshly or when removed from the bed owing to their original different bending and strengthening nature and loose binding. The stitch structure sufficiently compensates this fragility, as detailed in the following section.

Stitches to Enhance Binding Strength of Joints

As mentioned earlier, we added a stitch to connect each segment and enhance the binding strength between multiple materials. Creating a strong binding is significant to guarantee a seamless connection of each resource material segments, (1) making it easy to remove a printed filament from the print bed without breaking it, and (2) smoothly feed it to the nozzle when used for printing an object. We set the stitch length to 7 *mm*, to cover the top of both adjacent segments. We consider two types of stitch: (i) linear and (ii) zigzag. We tested using three materials in various combinations: colored-PLA, PVA, and conductive-PLA connected with a colored-PLA stitch. Figure 5 shows a comparison between two types of stitch in the resulting prints. We found that both stitches greatly enhance the connection, where the zigzag stitch presents a stronger link (i.e., Linear force presents 2N, 18.3N, 29.6N at breakage for PLA-PLA links, respectively).

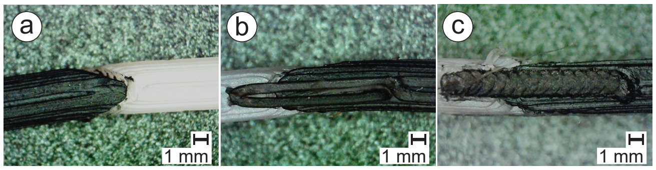


Figure 5. Comparison of different stitches to join two materials. The printed filament (a) without a stitch, and with (b) a line stitch and (c) a zigzag stitch.

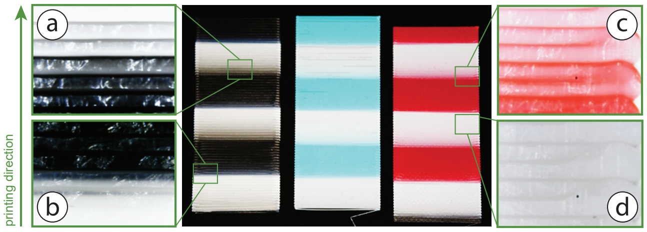


Figure 6. Printed sheets with gradation. In (a, c), the edges takes more lines to turn into the pure white (close to d), because the color residue remains in the nozzle. In comparison, (b) brighter color quickly switches to darker color.

Gradation Behavior

We observed a color gradation when switching between two segments takes place in the feeder. Technically, it is possible to mix two materials and obtain properties as a composite material. For example, at the point at which a material switch occurs, we can imagine that PLA mixes with TPU and might present properties similar to those of Nylon. However, special material properties such as flexibility, conductivity, and solvability cannot be easily observed because of external confounding factors. Thus, we only focus on the *color gradation* behavior from a standard PLA material in this experiment.

We first created a G-code to print a multi-colored filament in special dimensions within a flat sheet, as shown in Figure 6, to clearly observe a linear gradation. We specified the dimensions: a layer height to be 0.40 *mm* in a single-layered sheet and a width to be 30.07 *t-mm*, where each line in the sheet requires approximately 2.00 *e-mm* of filament to be printed. We arranged 150 horizontal lines in parallel to form a square sheet vertically. The width of each line is 0.48 *mm* (i.e., from calibration, $W_{nozzle} \times 120\%$ considering the occurrence of swelling [27]), which is sufficiently sparse to visually examine each line using a microscope. We prepared a printed filament consisting of transparent PLA and color-PLA materials to print this sheet. The transparent PLA is used for the brim and stitch, and the color-PLA materials form each segment (i.e., a striped pattern printed on the surface of the sheet).

As shown in Figure 6, the lighter material is easily overridden by switching to a darker material, while the opposite requires more steps to remove any residue. This can be easily observed in the white—black and white—red combinations. Figure 6a (black to white) and c (red to white) show that the edge of each line has difficulty turning into a pure white as shown in Figure 6d, owing to the residue color in the nozzle. Figure 6b (white to black) shows that color changing can happen in just one horizontal line. Counting the lines consisting of two colors allows us to find that approximately 5–12 lines are needed to clearly change two materials, depending on the color mixture. Because each line is printed with approximately 2.00 *t-mm* of filament, this means approximately 10–24 *e-mm* of filament is needed to change the colors without a gradation.

This length is a significant factor to be considered in implementing an end-to-end design system to facilitate the use of a printed filament. For example, if a user wants to hide the gradation effects into the infill region in the printed object, we can increase the infill density or the wall thickness to spend the mixed material parts. If the scale of a target 3D model does not allow this transition, a separate prime tower needs to be printed to budget the space, similar to the process used in existing dual color printing.

SYSTEM

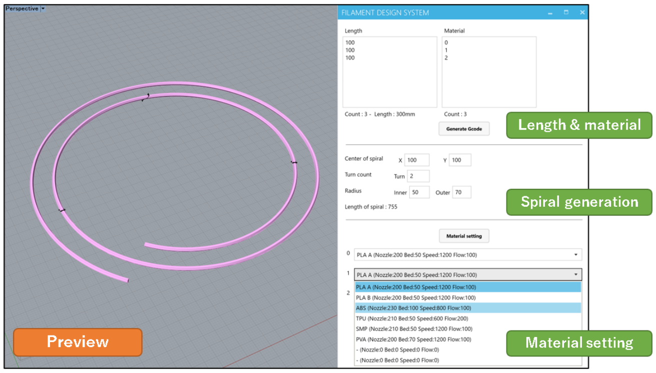
Integrating the empirical results found above, we implemented the design systems to support an end-to-end pipeline. Our system consists of two functional components, namely, filament design software to obtain the printed filaments, and three different modeling software with various input types *(i)* G-code, *(ii)* 3D model designed for dual printing, *(iii)* 2D image to obtain the parameters to be used as input in the filament design software. These software were implemented to provide various options for end-users, utilizing our programmable filament technique in different ways to offer unique design space, using *Rhinoceros* and *Grasshopper*, and the GUI using *HumanUI* library.

Filament Design Software

The purpose of this tool is to generate G-codes for printing a filament, taking a list of materials and the lengths of each segment as user input, as well as a list of material-specific slicing parameters corresponding to each segment.

User Interface and Workflow

The user interface consists of three sections as shown in Figure 7: a list of segment lengths representing each material in sequence, geometric specifications used to construct a spiral, and material-specific settings (e.g., print temperature, print speed). First, a user inputs the length of each segment to be spliced in order, in the first input box. Each time the user adds a new material in a row, a new drop-down menu specifying the material-specific slicing parameters is added to the material settings section. These segments constitute a spiral in a user-specified dimension. A spiral requires the center position, number of turns, and inner/outer radii. Technically, it is possible to generate a longer spiral than the total length of the segments; however, if the bed of the 3D printer is too small, it is necessary to adjust the specified radii accordingly. Then the user selects per-material settings from the system-generated drop-down menus. For example, if the user inputs $material = 0,1,2$ into the top input box, the software creates three drop-down lists in the material settings section at the bottom. Although the default settings show the recommended value of each material for best practice, the user can further modify the settings if necessary (e.g., increase the printing temperature). Finally, the system generates a ready-to-print G-code file by clicking the “Generate” button, which we detail the internal mechanism in the following section.



Length: 100 100 100

Material: 0 1 2

Count : 3 - Length : 300mm

Count : 3

Center of spiral X 100 Y 100

Turn count Turn 2

Radius Inner 50 Outer 70

Length of spiral : 755

0 PLA A (Nozzle:200 Bed:50 Speed:1200 Flow:100)

1 PLA A (Nozzle:200 Bed:50 Speed:1200 Flow:100)

2

- PLA A (Nozzle:200 Bed:50 Speed:1200 Flow:100)
- PLA B (Nozzle:200 Bed:50 Speed:1200 Flow:100)
- ABS (Nozzle:230 Bed:100 Speed:800 Flow:100)
- TPU (Nozzle:210 Bed:50 Speed:600 Flow:200)
- SMP (Nozzle:210 Bed:50 Speed:1200 Flow:100)
- PVA (Nozzle:200 Bed:70 Speed:1200 Flow:100)
- - (Nozzle:0 Bed:0 Speed:0 Flow:0)
- - (Nozzle:0 Bed:0 Speed:0 Flow:0)

Figure 7. Overview of the filament design software with interface window. Users design a printable filament using the input boxes (right), and the resultant filament can be previewed (left).

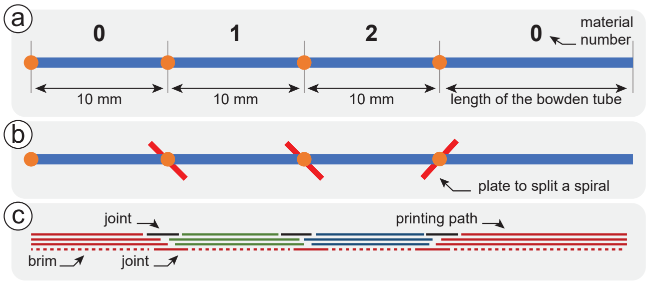


Figure 8. Process of the filament generation, with example input of $length = 10,10,10$ and $material = 0,1,2$. The software generates a spiral then (a) labels each segment, (b) splits the spiral into segments using cutting-planes, and (c) generates a G-code of the filament for each layer.

G-code generation Mechanism

We walkthrough the G-code generation process with the example shown in Figure 8, with lists of $length = 10,10,10$ and $material = 0,1,2$ as user input. The system generates a spiral given the geometric dimensions, and each segment is labeled according to the list of materials specified (Figure 8a). Note that the last segment is equal to the length of a Bowden tube to ease filament feeding, and is labeled to ’0’ matching with the very first segment. In our setting, the length of the spiral is 50 *mm* longer (the length of our Bowden tube) than the sum of each segment length.

Next, the system generates a wavy slicing-plane to split the spiral into segments (Figure 8b). The direction of the tilt of each slicing-plane is determined by each segment’s printing order, based on the material label. In the example, the first and final segments (labeled 0) are printed first, followed by the printing of segment 1. Therefore, a slicing-plane between segments 0 and 1 tilts toward segment 0, securing the printability. In the same way, a slicing-plane between segments 1 and 2 tilts toward segment 1, and as a result, segment 1 has two different types of cut at each end towards segment 0 and 2. Using the complete set of these slicing-planes, the system splits the spiral into a series of segments. To obtain a printing path of each segment, the system calculates an intersection between each segment and XY-planes. The XY-planes lined up in the z-direction with intervals of the layer height (we empirically set the layer height to 0.16 *mm*), and each intersection makes the contour of the segments.

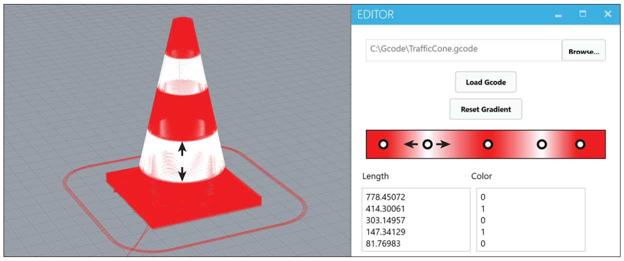
Finally, the software generates the G-code from the printing path. A series of commands that will be used to print each segment in this G-code are placed in order of the label, and when swapped to the next label, a pause command is inserted in between: moving to the XY-home position, unloading the current material, beeping to alert (M300), and pausing (M0). Commands to load and purge a new material (about 50-mm) follow. A command to print a short line at the corner to prime the nozzle is added, assuming resumed by the user after manual exchange of the source filament. When generating the G-code, we add brims around the first layer and stitches on the top and bottom of every segment across the entire spiral, to seamlessly join them (Figure 8c).

Material Settings

Each material comes with unique parameters: (i) type, (ii) nozzle temperature [°C], (iii) bed temperature [°C], (iv) printing speed [mm/min], and (v) flow rate [%]. Filament type (i) is referred to as an argument to call the pause command (M0). The LCD display of the 3D printer can show a custom message (e.g., 'Next: PLA') for the user's preference. Here, the temperatures are adjusted to fit the optimal conditions based on the properties of each material switched at a specific moment using the (ii) M109 command for the nozzle with the obtained parameters and (iii) M190 for the bed. The printing speed (iv) and flow rate (v) are accommodated using the G1 commands, with the F value applied to replace the printing speed, and using the E value multiplied by the given flow rate.

Modeling software #1: Reinterpreting G-code in Stripes

It is hard for users to know what specifications are required to print a needed filament. To support users who want to start with an input 3D model and are already familiar with existing slicing software, we provide a post-processor that generates a filament from pre-generated G-gode using *Cura* (Figure 9). As an example of post-process, this software allows users apply a stripe pattern to the G-code. First, a user imports a G-code file generated using Cura into the software, but any slicers can be used. We set the extrusion mode to *relative* in the slicer to easily calculate the amount of required material by accumulating the E values. The software interface shows a slider to add the material information to the input G-code. The left end of the slider indicates the bottom of the 3D model, and the right end corresponds to the top. The user can add anchor points to the slider by clicking on a blank space and define a color. By sliding the anchor points, a striped pattern in various lengths can be designed. Using the information in the slider, the software computes the amount of material needed to print segments in selected colors, and generates a list of parameters (segment lengths and material labels). After designing the striped pattern, users copy and paste the lists of parameters into the input box in the filament design interface above to obtain a G-code to print a filament.



C:\Gcode\TrafficCone.gcode

Length	Color
778.45072	0
414.30061	1
303.14957	0
147.34129	1
81.76983	0

Figure 9. The modeling *software #1* generates a list of required parameters to input into the filament design software, using an input G-code with slider to add colors.

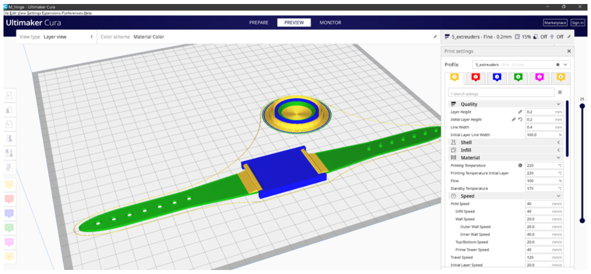


Figure 10. Cura slicer with our virtual 3D printer profile. Our settings define multiple extruders to generate a G-code with various materials dedicated by each extruder.

Modeling software #2: Slicing with Virtual Print Headers

Often, users want to print a 3D model originally designed for dual-color printing, easily obtainable from numerous free online repositories (e.g., *Thingiverse* and *Yeggie*). To support this scenario, we provide software enabling users to start with 3D models to specify the parameters required to attain a dedicated filament. First, we create a 3D printer profile (i.e., setting the building volume, extruder count, etc.) in Cura. This enables us to define a virtual 3D printer with multiple extruders account for each material spool assigned to them. For example, if a user has a 3D model that requires three different materials but owns a single-nozzle printer, the user can begin with the Cura profile using a virtual 3D printer with three headers. Note that we confirmed that Cura can add up to ten or more extruders regardless of the actual physical printer corresponding to this setting. Here, the user imports a watch model consisting of three parts; flexible bands, rigid joints, and a rigid display in three colors (Figure 10), and merged into one for alignment. Each part is assigned to each header with unique material settings. Slicing the model under this setting spits out a G-code that assumes it will be used in a physical tri-header printer, with the materials plugged accordingly.

The modeling software imports this G-code and calculates the amount of material needed in each extruder. This can be easily achieved by reading G-code commands T0, T1, ... Tn (where *n* indicates the extruder number) to switch between materials. The software captures them while calculating the amount of each material that will be used to 3D print the model and generates the lists of length and material settings to be used in the filament design software. In addition, to reuse this G-code in actual 3D printing the input 3D model using the printed filament, the system modifies the original G-code by removing Tn commands. The system yet leaves other material-specific commands such as temperature adjustment, as this

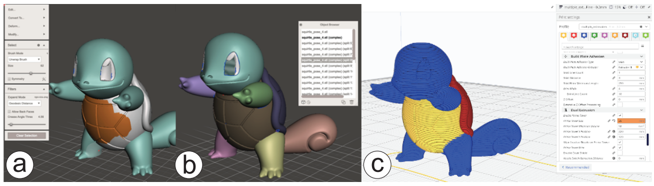


Figure 11. Users can use existing CAD tools (a) to select a 3D model in parts by brushing the surface, (b) automatically offsetting to segment into discrete volumes to export, (c) then import parts into Cura to obtain a list of parameters needed to print a filament.

is still required in making use of the printed multi-material filament in 3D printing. For example, if a single nozzle is fed with a TPU segment from the end of a PLA segment, the machine needs to wait until heating the nozzle for flexible materials (200 to 230+ °C). For 3D models needs support in other materials such as PVA, users can also assign one virtual header dedicated for supporting material only.

If a user does not own a 3D model with parts segmented for multi-material printing, several existing CAD tools (e.g., MeshMixer) provide a step-by-step guideline to segment one solid mesh into parts by offsetting [43], as shown in Figure 11.

Modeling software #3: Path-planning using a 2D Image

In addition to *software #2* that use existing 3D models, we further introduce software that takes a delicate 2D image as input then produce an appropriate printing path for building a facade. Here, we provide a software that supports converting 2D image to plan 3D-printing paths, by dividing discrete colors to appear exterior walls, while hiding gradation behavior inside of them (Figure 12). It does not require a prime tower outward, as we use this volume in mixed colors to print the so-called infill. With this structure, the system generates a printing path having a color change point inside the object such that the gradient effect does not appear outward. G-code generated by this software consists of thick walls with several layers in the cross sections. First, the system generates the walls and divides the outermost path on the object's facade according to an input image as shown in Figure 12. In this example, the outermost path is divided into three segments according to the number of colors appeared in the input image. After generating a path with the first material (red), the system connects the path to an inner wall to switch the materials (red to green). The length of the inner path is determined based on the expected gradation effect of two materials. Empirically, we set the default length to use a 20 *mm* material. When switching is complete to the next color, the path comes back to the facade (green), and the process iterates (green to red). By repeating this process according to the number of materials in a layer, a clear pattern appears on the facade without mixed colors.

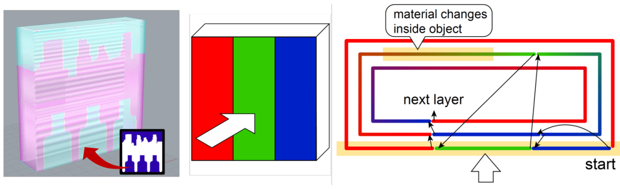


Figure 12. Our modeling *software #3* generates a unique path to utilize the inner structure of the 3D model and print the transition of the switching materials (center, right). Therefore, the pattern generated using an input image can appear in front of the wall (left).

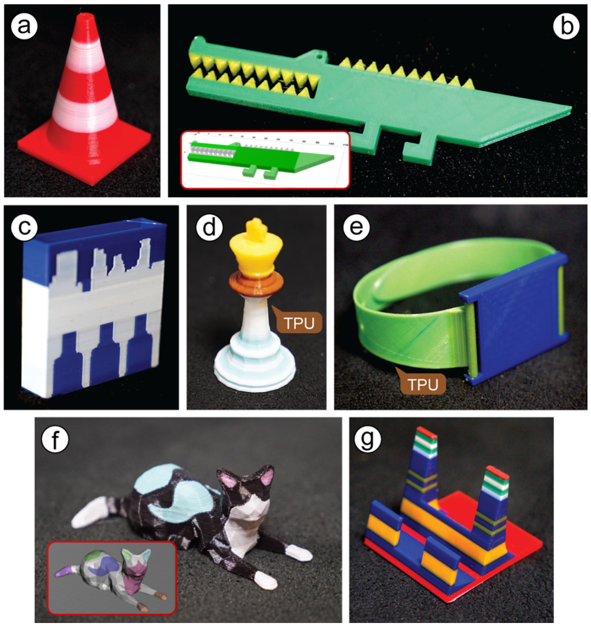


Figure 13. Example prints using our system: (a), (d) are created by *software #1*, (b), (e) and (f) are created by *software #2* with insets showing original input 3D models, and (c), (g) are printed by *software #3*.

Table 1 summarizes the unique design space, which enables users to choose from different input and explore design parameters to create multi-color/material.

Software	Input	Design space
#1	Gcode	Height of vertical segment
#2	3D model	Number of materials (head count)
#3	2D image	Facade dimension

Table 1. Three design software provide unique design benefit for end users, offering various options to choose from different input to design parameters in different design space.

VALIDATION WITH APPLICATION EXAMPLES

To validate the system's general use cases using various modeling methods (software) and materials throughout our end-to-end pipeline, we created a number of example applications. Figure 13 showcases these examples with the details summarized in Table 2. We replicated existing 3D models created for multi-material printing using a dual extruder, such as a popular 3D model used to test dual nozzle printing (Figure 13a), and an application of prior work (Figure 13b) using our pipeline. Figures 13d and 13g show that our technique is capable of printing more than five different materials, which is hard to achieve even with hardware upgrades— as the commercial, the latest multi-extruder machine comes with four extruders at most. Our technique also supports 3D models designed using common 3D modeling tools, such as *CraftML* [42] (Figure 13b), which makes it easy to design modular parts and export them using selective tags similar to jQuery, *ThinkerCAD* (Figure 13e), as well as *MeshMixer* (Figure 13f) that help users

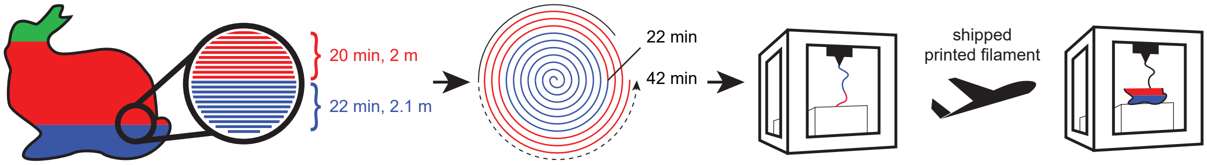


Figure 14. Upon the user's order of a 3D model with the desired materials, a custom filament can be manufactured on-demand and shipped to the user.

Name	Print time (filament)	Print time* (object)	Materials
Boat (Figure 1)	5h:4m:32s	1h:50m:17s	5 PLAs
Traffic cone (Figure 13a)	1h:12m:54s	51m:27s	2 PLAs
Alligator (Figure 13b)	1h:27m:53s	55m:7s	2 PLAs
UIST2020 logo (Figure 13c)	5h:5m:43s	2h:44m:55s	2 PLAs
Chessman (Figure 13d)	1h:33m:3s	58m:56s	4 PLAs TPU
Watch (Figure 13e)	3h:49m:1s	1h:48:17s	2 PLAs TPU
Cat (Figure 13f)	6h:5m:37s	2h:16m:28s	4 PLAs
Cellphone stands (Figure 13g)	2h:18m:26s	1h:28m:52s	5 PLAs

* We decreased printing speed to between 50-80% from the actual printing time (i.e., at 100% printing speed) to steady the printing process.

Table 2. Details of printed examples (the printing time was estimated from *Repetier-host* software).

align parts and export in place to merge for slicing. Figure 13c demonstrates sample prints made using the *software #3* for printing without an external prime tower. Purged materials during the color transition are used to print inside (replacing infill) the 3D printed object, reducing the printing time and material cost.

DISCUSSION & OUTLOOK

Post-processing vs. Pre-processing

Our technique flips the process to obtain a full color 3D printed object using a commodity 3D printer. Users often take a *post-processing* approach, cautiously brushing and painting the 3D printed result. By contrast, we propose a *pre-processing* approach, preparing a filament for multi-material printing in advance, and leaving the printer to take over. Printing a single threaded, multi-segmented filament still requires manually exchanging the resource filaments when switching the color and/or material. However, this process significantly reduces the number of manual filament exchanges required during conventional printing when using a single extruder printer [20]. The current layer-by-layer printing nature of an FDM 3D printer requires lots of numbers of exchange of materials even in one layer. For example, the boat shown in Figure 1 is printed using five different materials, requires only four exchanges (equal to the number of materials minus 1); however, if printed using a regular approach by manually exchanging the material every time a layer meets the new color segment, 211 exchanges are required, which is the number of extruder exchanges (Tn) appear in G-code—not ideal in practice.

On-demand Filament Manufacturing

Utilizing a *pre-processing* approach may, therefore, empower manufacturers in the future to produce a filament on-demand, opening a door for *customizable* materials. For example, upon a user's request to customize filament to print a 3D model using various materials, this on-demand filament that embeds the required filament properties, such as colors and location where these colors appear, can be ordered, manufactured, and finally shipped, as shown in Figure 14. Currently, materials are frequently purchased in bulk, from a single color batch, making it difficult for individuals to be equipped with full ranges of colors without purchasing 20+ filament spools for example. Our technique would unshackle users from budget issues in testing various colors and materials at low-investment. Our systems could serve as a first-class interface for the ordering system. In this future filament supply-chain, the customer and manufacturer can also closely work together, empowering the potential for manufacturers to become aware of the emerging needs in producing new materials for mass manufacturing.

Sharing of Data along with the 3D Model

The above speculative future is based on the possibility that users can share *data* describing the material requirements, as well as the 3D model. Using our pipeline, users can easily create a G-code to produce a special filament dedicated to printing a desired 3D model in multiple parts. In addition, once the user obtains the necessary input into the filament design system as the parameters (e.g., the type of materials, sequence, and length) that are determined by utilizing any of our modeling software (#1-#3), the user can simply share this model-specific data with others who can input the data into the filament design system, to replicate the filament with some variations in color or materials if needed.

Programming of Properties and Alternative Production

As introduced earlier, our technique also has the potential to produce materials with new characteristics, enabled by its flexibility in blending the materials. Current way of concatenating materials segments by segments could be laborious. Although mass-manufacturers in our visionary scenario can do a better job at concatenating, an FDM technique presents unique, wider opportunities in programmability, as presented in Figure 15. Our experiments demonstrated that a 3D printer can extrude two or more materials at the same time while mixing them, allowing the extruded material to become a new material. As shown in Figure 15 a-c, two materials are synthesized into a printed filament in various color ratios at each segment, exhibiting a gradual color gradation in the printed 3D object. Applying diverse composition methods of each segment, e.g., enclosing with another material (Figure 15c), yields new filament properties such as strength [19]. Similarly, the same

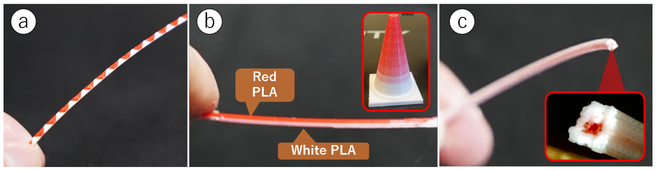


Figure 15. Various filaments printing techniques producing new properties. Filaments can be printed with (a) tiny segments, (b) a segment in which one material is layered on the other in varying portions resulting in gradient colors, and (c) a white filament with a red material inside.

phenomenon can emerge for different material combinations such as PLA + TPU [5] or PLA + Polycarbonate [6], if the material is mixed with a controllable portion of two materials to reproduce a programmable stiffness. The next step of this work includes exploring composite material production techniques in filament generation using an FDM technique, developing a prediction model to obtain programmable mechanical structures computed by mixture ratio upon quantified strength, then can be interpreted into Gcode for production.

LIMITATIONS & FUTURE WORK

Extra Hardware Configuration for Some Printers

During the process of printing a filament, we added z-hop by adding G1 Z commands to lift, to avoid a collision between the nozzle and the printed segments along the movement pathway, then lower the nozzle back to z-0 to print the next segment on the bed (see Figure 2b). However, some 3D printers have extra hardware that touch and drag the printed segment in the next spiral inward/outward even with this collision-proof technique, such as a magnetic end-stop to detect the proximity of a print-bed (e.g., in *Printrbot* and *Prusa Teacup*) or a cooling fan around the nozzle. In such cases, a special calibration technique needs to be applied, i.e., removing a fan or physically adjusting the end-stop height. When the end-stop is adjusted to be slightly above the true home (Z0), an offsetting distance must be manually found to start the printing at lower than where it homed. This offset needs to be added to the default setting (e.g., adding *G1 Z-4* as z-offset in Repetier-host).

Manual Works and Metrics for Success Measure

Our technique relies on the initial calibration process of each 3D printer. However, it is possible that residue of the previous segment is not completely purged and thus remains in the nozzle, particularly when printing with an extraordinarily long filament. Because E-steps can be manipulated only up to the first decimal place in the software setting, external factors such as the stiffness of the material, the temperature of the heater, and the stepper motor form factor, cannot be perfectly controlled via calibration. Empirically, we found that the amount of extrusion may vary during each printing process in our experiments. Further, common printing failures might also occur, such as clogging of the extruder or an under/over-extrusion, resulting in a shifted location where the actual material was supposed to appear. Figure 16(left) shows some samples with errors, such as a pattern shifted out of alignment, where the color did not correctly appear. Figure 16(right) also shows a red color mixed into a yellow color owing to the insufficient material cleaning. To solve these calibration and extrusion problems, we currently inspect the whole printing process and adjust the amount of extrusion when needed. Although the performances of machines are expected to be continuously improved in the future, a monitoring system can significantly help confirm the state of extrusion and which material is being extruded, such as when using an *OctoPrint* [7]. Along the way, we will need to develop user-centered metrics to measure success qualitatively, as it is possible that users feel frustrated due to the higher chance of clogged nozzle or not satisfied with quality of final prints.

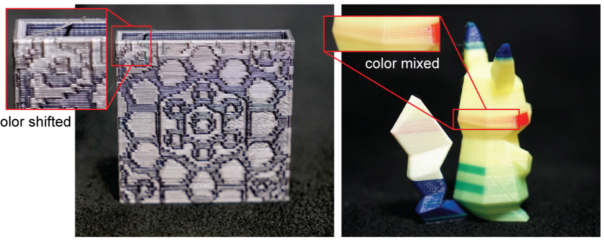


Figure 16. Failure cases of our method. Left: The color is shifted from a shift in the step motor feeding the material into the nozzle. Right: The red color is mixed into yellow owing to insufficient material cleaning during the printing process.

Validating Mechanical Properties and Material Range

From the experiment, we integrated empirical results into the system implementation with the best choice (e.g., zigzag stitch to blend two segments), but we have not gone through a full structural analysis for material composition and types. The nearest next step that will be followed by investigating programmable properties is conducting a holistic experiment to understand the mechanical behaviors of composite structure. For example, we will print segments (a) with a more variety of stitches (b) in various material compositions at each end and (c) in different lengths and the cross-sectional shapes (e.g., wavy cut and linear cut in varying angles), then test strength with two measures: (i) linear fracture by pulling two ends from each other in increasing forces, (ii) axial bending and breakage by adding growing weights at the seam. We expect to obtain the maximum strength of each material joints, and optimize structures upon input parameters of above variables that minimize laborious work needed in producing printed filaments. Further, we observed bigger differences in mechanical properties on each end result in easier breakage (e.g., TPU-PLA) in our initial experiment, thus, structural analysis of printed filaments with various materials will help us define appropriate range of materials to join, as well as recommend a better stitching strategy in each case.

CONCLUSION

In this paper, we proposed a programmable filament, a novel technique to 3D print multi-material objects using a low-cost FDM machine without any hardware modifications. We conducted a set of experiments to understand characteristics of the printed filament, then integrated these findings into designing an end-to-end fabrication pipeline. In this pipeline, we provide parametric design interfaces for end-users, used to design the filament directly, or by obtaining the specifications of a printable filament from input 2D images, G-code, or 3D models. We validated our approach is replicable with various types of machines and materials, then showcased printed examples. Finally, we discussed remaining technical challenges and a potential future achievable by the proposed technique.

ACKNOWLEDGMENTS

We thank Instructables user DasMia for the inspiration of this work, external reviewers for providing constructive feedback. This work was partially supported by Adobe Gift Award.

REFERENCES

[1] Davide Jose Nogueira Amorim, Troy Nachtigall, and Miguel Bruns Alonso. 2019. Exploring Mechanical Meta-Material Structures through Personalised Shoe Sole Design. In *Proceedings of the ACM Symposium on Computational Fabrication (SCF '19)*. Association for Computing Machinery, New York, NY, USA, Article Article 2, 8 pages. DOI: http://dx.doi.org/10.1145/3328939.3329001

[2] Byoungkwon An, Ye Tao, Jianzhe Gu, Tingyu Cheng, Xiang "Anthony" Chen, Xiaoxiao Zhang, Wei Zhao, Youngwook Do, Shigeo Takahashi, Hsiang-Yun Wu, Teng Zhang, and Lining Yao. 2018. Thermorph: Democratizing 4D Printing of Self-Folding Materials and Interfaces. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. Association for Computing Machinery, New York, NY, USA, Article Paper 260, 12 pages. DOI: http://dx.doi.org/10.1145/3173574.3173834

[3] DasMia. 2019. DIY Rainbow 3D Printing Filament. (2019). https://www.instructables.com/id/DIY-Rainbow-3D-Printing-Filament/

[4] Daniel Groeger, Martin Feick, Anusha Withana, and Jürgen Steimle. 2019. Tactlets: Adding Tactile Feedback to 3D Objects Using Custom Printed Controls. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19)*. Association for Computing Machinery, New York, NY, USA, 923–936. DOI: http://dx.doi.org/10.1145/3332165.3347937

[5] Yi Guo, Lei Yan, Zhu Zeng, Lin Chen, Mingxue Ma, Rui Luo, Jun Bian, Hailan Lin, and Daiqiang Chen. 2019. TPU/PLA nanocomposites with improved mechanical and shape memory properties fabricated via phase morphology control and incorporation of multi-walled carbon nanotubes nanofillers. *Polymer Engineering & Science* (2019). DOI: http://dx.doi.org/10.1002/pen.25365

[6] Kevin R. Hart, Ryan M. Dunn, and Eric D. Wetzel. 2020. Tough, Additively Manufactured Structures Fabricated with Dual-Thermoplastic Filaments. *Advanced Engineering Materials* 22, 4 (2020), 2070013. DOI: http://dx.doi.org/10.1002/adem.202070013

[7] Gina Häußge. 2020. OctoPrint. (2020). https://octoprint.org/

[8] Jean Hergel and Sylvain Lefebvre. 2014. Clean Color: Improving Multi-Filament 3D Prints. *Comput. Graph. Forum* 33, 2 (May 2014), 469–478. DOI: http://dx.doi.org/10.1111/cgf.12318

[9] Alexandra Ion, Robert Kovacs, Oliver S. Schneider, Pedro Lopes, and Patrick Baudisch. 2018. Metamaterial Textures. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. Association for Computing Machinery, New York, NY, USA, Article Paper 336, 12 pages. DOI: http://dx.doi.org/10.1145/3173574.3173910

[10] Vikram Iyer, Justin Chan, Ian Culhane, Jennifer Mankoff, and Shyamnath Gollakota. 2018. Wireless Analytics for 3D Printed Objects. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology (UIST '18)*. Association for Computing Machinery, New York, NY, USA, 141–152. DOI:http://dx.doi.org/10.1145/3242587.3242639

[11] Vikram Iyer, Justin Chan, and Shyamnath Gollakota. 2017. 3D Printing Wireless Connected Objects. *ACM Trans. Graph.* 36, 6, Article Article 242 (Nov. 2017), 13 pages. DOI:http://dx.doi.org/10.1145/3130800.3130822

[12] Yuhua Jin, Isabel Qamar, Michael Wessely, Aradhana Adhikari, Katarina Bulovic, Parinya Punpongsanon, and Stefanie Mueller. 2019. Photo-Chromeleon: Re-Programmable Multi-Color Textures Using Photochromic Dyes. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19)*. Association for Computing Machinery, New York, NY, USA, 701–712. DOI: http://dx.doi.org/10.1145/3332165.3347905

[13] Jeeeun Kim, Clement Zheng, Haruki Takahashi, Mark D Gross, Daniel Ashbrook, and Tom Yeh. 2018. Compositional 3D Printing: Expanding & Supporting Workflows Towards Continuous Fabrication. In *Proceedings of the 2nd ACM Symposium on Computational Fabrication (SCF '18)*. ACM, New York, NY, USA, Article 5, 10 pages. DOI: http://dx.doi.org/10.1145/3213512.3213518

[14] Tyler Koslow. 2016. Pattette2S. (2016). https://www.mosaicmfg.com/products/palette-2s

[15] Tyler Koslow. 2019. New Multi-Material Upgrade Released For Prusa i3 MK2 3D Printer. (2019). https://3dprint.com/150853/multi-material-prusa-upgrade/

[16] Tim Kuipers, Eugeni Doubrovski, and Jouke Verlinden. 2017. 3D Hatching: Linear Halftoning for Dual Extrusion Fused Deposition Modeling. In *Proceedings of the 1st Annual ACM Symposium on Computational Fabrication (SCF '17)*. ACM, New York, NY, USA, Article 2, 7 pages. DOI: http://dx.doi.org/10.1145/3083157.3083163

[17] Gierad Laput, Xiang 'Anthony' Chen, and Chris Harrison. 2015. 3D Printed Hair: Fused Deposition Modeling of Soft Strands, Fibers, and Bristles. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 593–597. DOI: http://dx.doi.org/10.1145/2807442.2807484

[18] Dingzeyu Li, Avinash S. Nair, Shree K. Nayar, and Changxi Zheng. 2017. AirCode: Unobtrusive Physical Tags for Digital Fabrication. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17)*. Association for Computing Machinery, New York, NY, USA, 449–460. DOI: http://dx.doi.org/10.1145/3126594.3126635

[19] Tiantian Li, Yanyu Chen, and Lifeng Wang. 2018. Enhanced fracture toughness in architected interpenetrating phase composites by 3D printing. *Composites Science and Technology* 167 (2018), 251 – 259. DOI: http://dx.doi.org/10.1016/j.compscitech.2018.08.009

[20] Computer Aided Technology LLC. 2019. Changing Material Colors for FDM 3D Printers | A Tip from a Tech. (2019). https://www.cati.com/blog/2019/11/changing-material-colors-for-fdm-3d-printers-a-tip-from-a-tech/.

[21] Henrique Teles Maia, Dingzeyu Li, Yuan Yang, and Changxi Zheng. 2019. LayerCode: Optical Barcodes for 3D Printed Shapes. *ACM Trans. Graph.* 38, 4, Article Article 112 (July 2019), 14 pages. DOI: http://dx.doi.org/10.1145/3306346.3322960

[22] Stefanie Mueller, Sangha Im, Serafima Gurevich, Alexander Teibrich, Lisa Pfisterer, François Guimbretière, and Patrick Baudisch. 2014. WirePrint: 3D Printed Previews for Fast Prototyping. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 273–280. DOI: http://dx.doi.org/10.1145/2642918.2647359

[23] Martin Nisser, Junyi Zhu, Tianye Chen, Katarina Bulovic, Parinya Punpongsanon, and Stefanie Mueller. 2019. Sequential Support: 3D Printing Dissolvable Support Material for Time-Dependent Mechanisms. In *Proceedings of the Thirteenth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '19)*. Association for Computing Machinery, New York, NY, USA, 669–676. DOI: http://dx.doi.org/10.1145/3294109.3295630

[24] Parinya Punpongsanon, Xin Wen, David S. Kim, and Stefanie Mueller. 2018. ColorMod: Recoloring 3D Printed Objects Using Photochromic Inks. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. Association for Computing Machinery, New York, NY, USA, Article Paper 213, 12 pages. DOI: http://dx.doi.org/10.1145/3173574.3173787

[25] Tim Reiner, Nathan Carr, Radomír Měch, Ondřej Šťava, Carsten Dachsbacher, and Gavin Miller. 2014. Dual-color Mixing for Fused Deposition Modeling Printers. *Comput. Graph. Forum* 33, 2 (May 2014), 479–486. DOI:http://dx.doi.org/10.1111/cgf.12319

[26] Dan Rock. 2019a. Extruder steps per mm. (2019). https://e3d-online.dozuki.com/Guide/Extruder+steps+per+mm/96

[27] Dan Rock. 2019b. Flow rate (Extrusion multiplier) calibration guide. (2019). https://e3d-online.dozuki.com/Guide/Flow+rate+(Extrusion+multiplier)+calibration+guide./89

[28] H. Tanaka S. Koda. 2017. Direct G-code manipulation for 3D material weaving. In *Proceedings of SPIE Smart Structures and Materials + Nondestructive Evaluation and Health Monitoring*, Vol. 10167. DOI: http://dx.doi.org/10.1117/12.2261648

[29] Martin Schmitz, Mohammadreza Khalilbeigi, Matthias Balwierz, Roman Lissermann, Max Mühlhäuser, and Jürgen Steimle. 2015. Capricate: A Fabrication Pipeline to Design and 3D Print Capacitive Touch Sensors for Interactive Objects. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. Association for Computing Machinery, New York, NY, USA, 253–258. DOI: http://dx.doi.org/10.1145/2807442.2807503

[30] Liang Shi, Vahid Babaei, Changil Kim, Michael Foshey, Yuanming Hu, Pitchaya Sitthi-Amorn, Szymon Rusinkiewicz, and Wojciech Matusik. 2018. Deep Multispectral Painting Reproduction via Multi-Layer, Custom-Ink Printing. *ACM Trans. Graph.* 37, 6, Article Article 271 (Dec. 2018), 15 pages. DOI: http://dx.doi.org/10.1145/3272127.3275057

[31] Shopify. 2020. FilaBlend. (2020). https://www.filablend.com/

[32] Pitchaya Sitthi-Amorn, Javier E. Ramos, Yuwang Wangy, Joyce Kwan, Justin Lan, Wenshou Wang, and Wojciech Matusik. 2015. MultiFab: A Machine Vision Assisted Platform for Multi-Material 3D Printing. *ACM Trans. Graph.* 34, 4, Article Article 129 (July 2015), 11 pages. DOI:http://dx.doi.org/10.1145/2766962

[33] Mark A. Skylar-Scott, Jochen Mueller, Claas W. Visser, and Jennifer A. Lewis. 2019. Voxelated soft matter via multimaterial multinozzle 3D printing. *Nature* 575 (2019), 330–335. DOI: http://dx.doi.org/10.1038/s41586-019-1736-8

[34] Haichuan Song, Jonàs Martínez, Pierre Bedell, Noémie Vennin, and Sylvain Lefebvre. 2019. Colored Fused Filament Fabrication. *ACM Trans. Graph.* 38, 5, Article Article 141 (June 2019), 11 pages. DOI: http://dx.doi.org/10.1145/3183793

[35] Stratasys. 2020. Stratasys Connex3, Objet500 and Objet350. (2020). https://www.stratasys.com/3d-printers/objet-350-500-connex3

[36] Haruki Takahashi and Jeeeun Kim. 2019. 3D Printed Fabric: Techniques for Design and 3D Weaving Programmable Textiles. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19)*. Association for Computing Machinery, New York, NY, USA, 43–51. DOI: http://dx.doi.org/10.1145/3332165.3347896

[37] Haruki Takahashi and Homei Miyashita. 2017. Expressive Fused Deposition Modeling by Controlling Extruder Height and Extrusion Amount. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 5065–5074. DOI: `http://dx.doi.org/10.1145/3025453.3025933`

[38] Ryan L. Truby and Jennifer A. Lewis. 2016. Printing soft matter in three dimensions. *Nature* 540, 7633 (2016), 371–378. DOI: `http://dx.doi.org/10.1038/nature21003`

[39] Joshua Vasquez, Hannah Twigg-Smith, Jasper Tran O'Leary, and Nadya Peek. 2020. Jubilee: An Extensible Machine for Multi-Tool Fabrication. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. DOI:`http://dx.doi.org/10.1145/3313831.3376425`

[40] Kiril Vidimče, Szu-Po Wang, Jonathan Ragan-Kelley, and Wojciech Matusik. 2019. OpenFab: A Programmable Pipeline for Multimaterial Fabrication. *Commun. ACM* 62, 9 (Aug. 2019), 97–105. DOI: `http://dx.doi.org/10.1145/3344808`

[41] XYZprinting. 2020. XYZprinting da Vinch Color. (2020). `https://www.xyzprinting.com/en-US/product/da-vinci-color`

[42] Tom Yeh and Jeeeun Kim. 2018. CraftML: 3D Modeling is Web Programming. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. Association for Computing Machinery, New York, NY, USA, Article Paper 527, 12 pages. DOI: `http://dx.doi.org/10.1145/3173574.3174101`

[43] Jonny Yeu. 2019. Convert Single-Color STL files into Multi-Body Models using MeshMixer. (2019). `https://support.mosaicmfg.com/hc/en-us/articles/115002827014`

[44] Yang Zhang, Gierad Laput, and Chris Harrison. 2017. Electrick: Low-Cost Touch Sensing Using Electric Field Tomography. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. Association for Computing Machinery, New York, NY, USA, 1–14. DOI: `http://dx.doi.org/10.1145/3025453.3025842`